

# Microsoft® Small Basic

## Стеки и массивы

Предполагаемое время работы с этим  
уроком: 1 час



# Стеки и массивы

## В этом уроке вы изучите

Использование различных операций объекта **Array**.

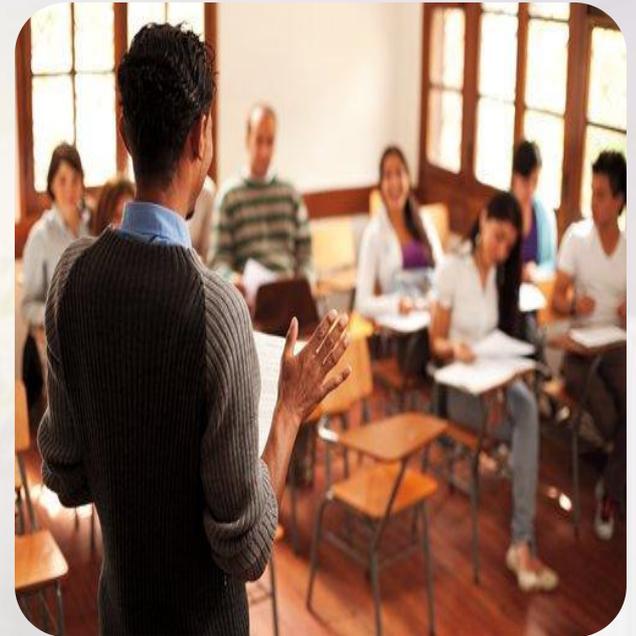
Использование различных операций объекта **Stack**.



# Стеки и массивы

Перед рассмотрением объектов **Array** и **Stack** сначала необходимо выяснить, как можно использовать эти объекты.

Массив может иметь множество размеров, но стек имеет только один размер. Можно напрямую обращаться к любому элементу в массиве, но можно обращаться только к верхнему элементу стека. Другими словами, для доступа к последнему элементу стека необходимо перебрать все его элементы.



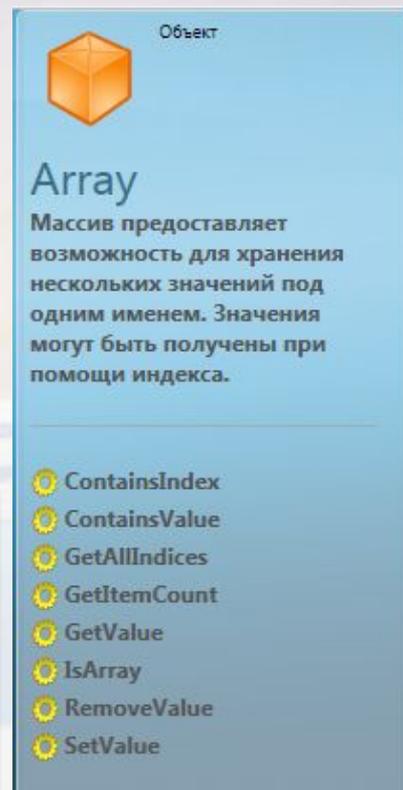
# Объект Array

Пока вы изучили переменные, в которых сохраняются отдельные значения. Теперь изучим особый вид переменной, которая называется массивом.

В массиве одновременно может храниться несколько значений. Если необходимо сохранить имена пяти пользователей, можно создать пять переменных или всего одну переменную для хранения всех пяти имен.

Для сохранения нескольких значений в массиве используется метод индексирования. Например, можно создать следующий массив с именем `name`: `name[1]`, `name[2]`, `name[3]`, `name[4]` и `name[5]`. Здесь 1, 2, 3, 4 и 5 — это индексы для имени массива.

Метки `name[1]`, `name[2]`... могут обозначать различные переменной, но все они представляют только одну переменную!



Объект



## Array

Массив предоставляет возможность для хранения нескольких значений под одним именем. Значения могут быть получены при помощи индекса.

- ContainsIndex
- ContainsValue
- GetAllIndices
- GetItemCount
- GetValue
- IsArray
- RemoveValue
- SetValue

# Операции объекта Array

Теперь рассмотрим некоторые операции объекта **Array**, такие как **isArray**, **containsIndex** и **containsValue**.

Можно определить, является ли указанная переменная массивом, с помощью операции **isArray**.

Можно определить, содержит ли массив указанный индекс, с помощью операции **containsIndex**. Эта операция полезна при необходимости определения того, инициализирует ли указанное значение индекс массива.

Можно определить, содержит ли массив указанное значение, с помощью операции **containsValue**. Эту операцию можно использовать для определения того, хранится ли значение массива в указанном индексе.

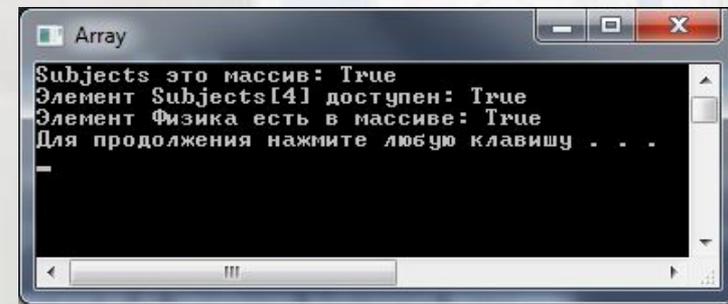
# Операции объекта Array

Посмотрим, как можно использовать эти операции в программе.

```
Subjects[1] = "Английский"  
Subjects[2] = "История"  
Subjects[3] = "Информатика"  
Subjects[4] = "Математика"  
Subjects[5] = "Физика"  
  
TextWindow.WriteLine("Subjects это массив: " + Array.IsArray(Subjects))  
TextWindow.WriteLine("Элемент Subjects[4] доступен: " + Array.ContainsIndex(Subjects, 4))  
TextWindow.WriteLine("Элемент Физика есть в массиве: " + Array.ContainsValue(Subjects, "Физика"))  
Array.GetItemCount(Subjects)
```

В этом примере в массиве **Предметы** хранятся названия пяти предметов. Можно проверить, является ли переменная **Предметы** массивом, используя операцию **IsArray**. Также можно проверить наличие индекса **Предметы[4]**, используя операцию **ContainsIndex**. Можно проверить наличие значения "Математика" в массиве **Предметы**, используя операцию **ContainsValue**.

**ВЫВОД**



```
Array  
Subjects это массив: True  
Элемент Subjects[4] доступен: True  
Элемент Физика есть в массиве: True  
Для продолжения нажмите любую клавишу . . .
```

# Операции объекта Array

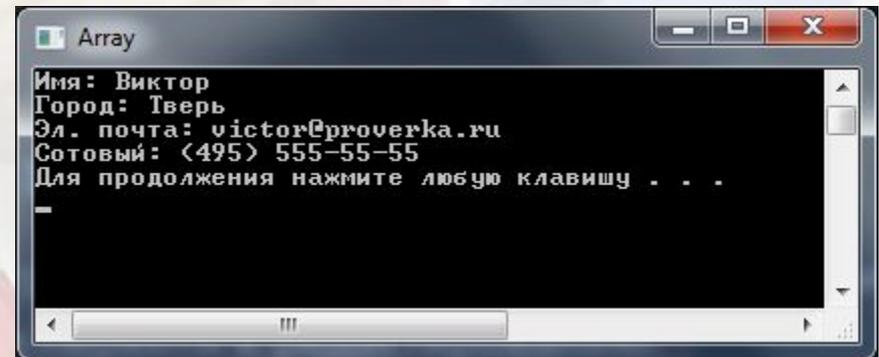
Объект **Array** также предоставляет другие полезные операции, такие как:

- **GetAllIndices**
- **GetItemCount**

Рассмотрим следующий пример, чтобы узнать, как использовать эти операции.

```
Employee["Имя"] = "Виктор"  
Employee["Город"] = "Тверь"  
Employee["Эл. почта"] = "victor@proverka.ru"  
Employee["Сотовый"] = "(495) 555-55-55"  
Emplist = Array.GetAllIndices(Employee)  
For i = 1 To Array.GetItemCount(Employee)  
    TextWindow.WriteLine(Emplist[i] + ": " + Employee[Emplist[i]])  
EndFor
```

В этом примере индексы массива **Сотрудники** неизвестны, поэтому используется операция **GetAllIndices**. Далее используется операция **GetItemCount** в цикле **For loop** для вывода списка информации, сохраненной в массиве **Сотрудники**.



```
Array  
Имя: Виктор  
Город: Тверь  
Эл. почта: victor@proverka.ru  
Сотовый: <495> 555-55-55  
Для продолжения нажмите любую клавишу . . .  
-
```

# Объект Stack

Объект **Stack** можно использовать для хранения данных наподобие стойки. Этот объект работает по принципу ЛИФО.

Например, если посмотреть на стек сверху, вы увидите только верхний элемент. Чтобы просмотреть следующий элемент, необходимо снять верхний. Элемент в середине стека можно просмотреть только после снятия всех верхних элементов.

Объект **Stack** состоит из трех операций.

**PushValue**

**PopValue**

**GetCount**

Рассмотрим каждую из этих операций...

# Операции объекта Stack

Объект **Stack** сохраняет данные наподобие стопки тарелок. Рассмотрим несколько примеров, чтобы понять работу этого объекта.

Использование операции **PushValue** подобно добавлению элемента на верх стойки. С помощью этой операции можно добавить значение в указанный стек.

```
Stack.PushValue(Stack1, 2)
```

Использование операции **PopValue** подобно снятию элемента с верха стойки. Эту операцию можно использовать для извлечения значения из указанного стека.

```
Stack.PopValue(Stack1)
```

Операция **GetCount** предоставляет общее число элементов в стеке. Эту операцию можно использовать для определения числа элементов в стеке.

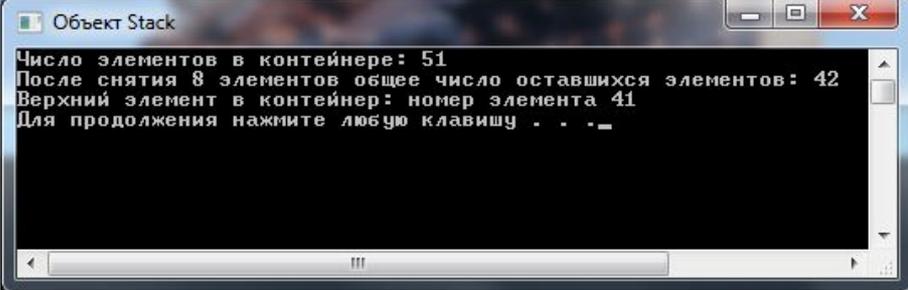
```
Stack.GetCount(Stack1)
```

# Операции объекта Stack

Напишем программу, чтобы лучше понять эти операции.

```
container = "пусто"  
For i = 0 To 50  
    Stack.PushValue(container, "номер элемента " + i)  
EndFor  
TextWindow.WriteLine("Число элементов в контейнере: " + Stack.GetCount(container))  
For i = 0 To 8  
    Stack.PopValue(container)  
EndFor  
TextWindow.WriteLine("После снятия 8 элементов общее число оставшихся элементов: " + Stack.GetCount(container))  
TextWindow.WriteLine("Верхний элемент в контейнер: " + Stack.PopValue(container))
```

В этом примере операция **PushValue** используется для добавления 50 элементов в пустой контейнер. Затем из стека извлекаются восемь элементов с помощью операции **PopValue**. После этого используется операция **GetCount** для получения числа оставшихся элементов. Можно отобразить значение верхнего элемента.



```
Объект Stack  
Число элементов в контейнере: 51  
После снятия 8 элементов общее число оставшихся элементов: 42  
Верхний элемент в контейнер: номер элемента 41  
Для продолжения нажмите любую клавишу . . .
```

# Подведем итоги...



**Поздравляем! Вы изучили**

- Использование различных операций объекта **Stack**.
- Использование различных операций объекта **Array**.

# Продемонстрируйте свои знания

**С помощью объекта Array можно написать программу бронирования авиабилетов, используемую для выполнения следующих действий.**

- ❖ Резервирование мест для 10 пассажиров.
- ❖ Отображение имени и номера места каждого пассажира.
- ❖ Отображение числа доступных мест.

